Taming Big Data Final Project Report

Yan (Stella) Si

May 9th, 2024

1 Abstract

This project investigates core-set clustering and its capabilities in comparison to full-set clustering. We applied these two methods on an annotated yelp review dataset to evaluate their performance. The objective of clustering is to accurately predict the sentiment of the reviews, of whether the tone is positive or negative. We attempted three word representation methods, and three partition methods for core-set selection. We can conclude that BERT fine-tuning word embedding performed the best during clustering, while no clustering accurately predicted polarity. We also found sufficient evidence that core-set clustering is a favourable alternative to full-set clustering for greater computational efficiency. The report delineats the methodology, the dataset, the results, and future direction.

2 Methodology

2.1 K-means Clustering

K-means clustering divides a number of data points into k clusters where points assigned to clusters are closest in their means. The algorithm aims to minimize the distance of the points to their centroids. The algorithm first selects a random set of k points and set them as the centroid. Then, it calculates the distance between the points to all centroids and assign them to a cluster. Once clusters are assigned, centroids are updated. The cluster assignment and centroid recalculation steps will repeat until convergence which is when centroids no longer change significantly. The algorithm then outputs the final centroids and its assigned data points.

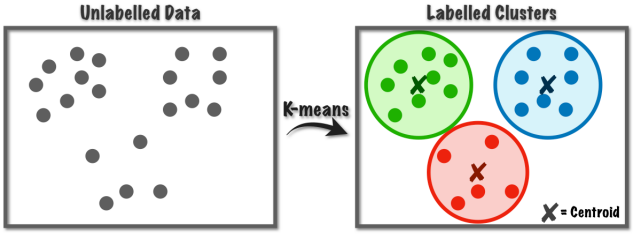


Figure 1: Illustration of k-means clustering[1]

2.2 Coreset Clustering

In core-set clustering, we decrease the size of the data set by selecting a smaller set of data to increase the efficiency of clustering. To do this, we partition the data set into n segments in different ways and sample data points from each portion. The resulting clustering data set would be the union of these samples. We will still use K-means clustering in the core-set data set.

2.3 Principal Component Analysis (PCA)

PCA is a linear dimensionality reduction technique that transforms a high dimension data set into a lower one while still containing most of the information from the original data set. The dimensionality of the text representation vectors are large. Therefore, we used PCA to cluster efficiently.

2.4 Silhouette Score

The silhouette score is used to evaluate the quality of the final clusters. The silhouette value, which varies from -1 to +1, reflects how similar an object is to its own cluster compared to other clusters. High value means that the point is more similar to its own cluster than other clusters, while a lower value means that the cluster does not work well for the assigned points. Here is the function of how the value is obtained.

$$s_i = \frac{b_i - a_i}{max(bi, ai)}$$

a_i is the average distance between i and all of the other points in its own cluster.

$$a_i = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

b_i is the average distance between i and its next nearest cluster centriod.

$$b_i = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

3 Real-world Dataset

The annotated yelp review data was used in a journal paper and posted on to Github[4]. The data set contained six variables: review text, opinion target, category, polarity, from (character position), and to (character position) [Table 1].

Review Text (N = 1734)	**Opinion Target**	**Category**
Reviews of restaurants in London	The fine grain topic of the review (e.g., waiter, wine list, fries, etc.)	Category of the reviews
Polarity	**From**	**To**
Whether the review is negative, positive, or neutral	Marking the begin and end point of character number where the polarity is identified	

Table 1: Detailed Description of Dataset Features

The main categories of the reviews consisted of restaurant, food, service, ambiance, location, and drinks, and there were subcategories within each category, consisting of 12 unique combinations of subcategories. For polarity, 63 percent of the reviews were positive, 12 percent of the reviews were negative, 2 percent of the reviews were neutral, while the rest did not have a polarity rating. For the sake of this analysis, we only kept the reviews that were positive or negative.

4 Data Processing

4.1 Text Cleaning

Review text cleaning were carried first to reduce noise during clustering. We converted the words to lowercase and eliminated the punctuation. We then removed stop words, which are common words

that do not add any meaning like “the”, “you”, “at”. Since the goal is to identify the polarity of review texts, we also removed words that are not related to sentiment such as the restaurant names, food names, and redundant relational words such as ’additionally’, ’across’ and ’next’. Other processing included removing abnormally small or large words to filter out grammar mistakes, words that were not all letters, and empty spaces and rows.

4.2 Tokenization and Word Embedding

Next, We tokenize the cleaned review text to smaller units to make it easier for the machine to process. For word embedding, which is the process of converting text into numerical representation vectors, we utilized three methods:

Figure 2: Illustration of tokenizer[2]

a. Term frequency-inverse document frequency (TF-IDF) with weights

This algorithm uses the term frequency score and inverse document frequency score to determine the relevant words in a text. The term frequency score measures the frequency of words within a piece of text. The inverse document frequency score captures the rarity of words in the text in relation to all pieces of text, giving rare words more weight. The TF-IDF vectors for the review data consists of the TF-IDF scores for each word calculated by the multiplication of the term freqency score and inverse document frequency score for each word.

$$\text{Term Frequency}_{ij} = \frac{\text{term i frequency in document j}}{\text{total number of terms in document j}}$$

$$\text{Inverse Document Frequency}_i = \log \frac{\text{total documents}}{\text{number of documents containing term i}}$$

$$\text{tf-idf}_{i,j} = \text{Term Frequency}_{ij} \times \text{Inverse Document Frequency}_i$$

Furthermore, since the objective of clustering is to find polarity, we added positive weights for words like “good” and “fantastic” and negative weights for words like “bad” and “terrible” and updated the TD-IDF vector representations.

b. Bidirectional Encoder Representation from Transformers (BERT) model - Feature based approach

Research shows thats that the BERT model can perform better text representation for clustering than TF-IDF as it takes the position and context of a word in a sentence into account[3]. The BERT language model is a deep learning model that reads the input text all at once (vs. sequentially) and trained on masked language modeling (MLM) and next sentence prediction (NSP) tasks. MLM training hides a word in a sentence and the algorithm predicts the masked word. NSP training has the algorithm predict whether two sentences have a connection between the two. Thus, it gives the model capacity to “understand” the text further than just the frequency of words within a text.

The BERT base model has 12 layers of transformer encoders each with a hidden size of 768. To get the word embedding, we are specifically interested in the hidden size of the last transformer layer, giving us the vector representation of the texts.

c. BERT fine-tuned model

To increase the ability of detecting polarity during clustering, we further fine-tuned the base BERT model with review labels of polarity. After training, we extracted the vector representation of the text from the hidden size of the last transformer layer from the fine-tuned model, which achieved a prediction accuracy of 0.93.

5 Algorithm Implementation

For full-set clustering, we use all three methods of word embedding to run k-means clustering, yielding three experimental results. We set k=2 as we are interested in polarity of positive and negative.

We selected the BERT fine-tuned word embedding to run core-set clustering on because it performed the best during full-set clustering. Three types of partitions were attempted to extract samples: random partition, review category partition, and polarity partition. Partitioning based on polarity is conceptually contradicting as polarity is the predicted outcome, but it is worth exploring.

For random partition, the dataset was divided randomly into six groups, and twenty percent of the data points were sampled from each group. For category partition, the dataset was divided into six main category groups: restaurant, food, service, ambiance, location, and drinks. Half of the samples were randomly selected from each group. For polarity partition, the dataset was divided into two groups of positive and negative sentiment, and a quarter of the samples were randomly selected from each group. These sampling proportions were determined to explore the range of core-set clustering performance. We will still use k-means clustering and set k=2 for best comparison.

6 Results

a. Silhouette score

The BERT fine-tuned word embedding performed the best during full-set clustering, yielding a silhouette score of 0.533, which is reasonable by clustering standards. The TD-IDF word embedding follows with a silhouette score of 0.306, and the BERT feature-based model performed the worst, which did not account for polarity in its encoding.

For core-set clustering, random and category partition clustering silhouette scores did not deteriorate a great deal from the full-set silhouette score of 0.533, suggesting robust word embedding and acceptability of core-set clustering. The polarity partition core clustering yielded a slightly better silhouette score from the original clustering, possibly due to the recursive partition and cluster.

TD-IDF	**BERT Feature-based**	**BERT Fine-tuned**
0.306	0.172	0.533
Random Partition	**Category Partition**	**Polarity Partition**
0.522	0.523	0.539

Table 2: Silhouette score of the clustering algorithms

b. Top words and proportion of positive and negative reviews in clusters

TD-IDF top words:
Cluster 1: excellent, service, staff, value, dinner, atmosphere, money, quality, good, always
Cluster 2: good, service, great, staff, place, excellent, always, atmosphere, table, london

BERT Feature-based:
Cluster 1: good, service, great, excellent, staff, place, wine, always, menu, atmosphere
Cluster 2: good, great, service, excellent, place, would, lovely, staff, always, lunch

BERT Fine-tuned:
Cluster 1:good, service, great, excellent, staff, place, lovely, table, menu, always
Cluster 2:good, service, great, excellent, staff, place, always, london, atmosphere, one

Random Partition:
Cluster 1: good, service, excellent, great, place, staff, always, lovely, experience, time
Cluster 2: good, great, service, excellent, table, water, lunch, wine, love, every

Category Partition:
Cluster 1: good, great, service, excellent, lovely, staff, place, menu, one, love
Cluster 2:good, service, great, excellent, staff, always, place, wine, atmosphere, one

Polarity Partition:
Cluster 1: good, great, service, excellent, place, staff, always, atmosphere, table, really
Cluster 2: service, good, great, excellent, place, well, menu, atmosphere, best, wine

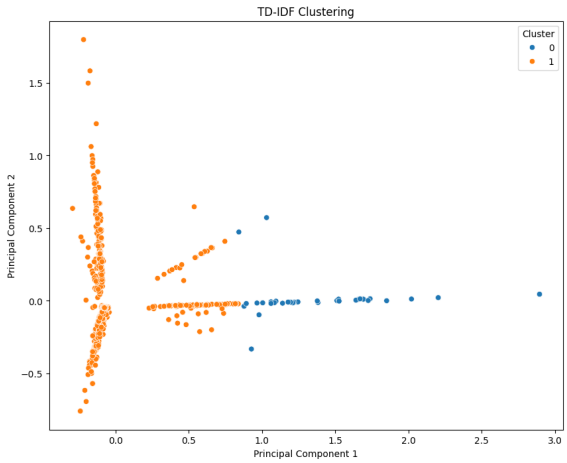
Cluster	Cluster 1	Cluster 2	Cluster 1	Cluster 2	Cluster 1	Cluster 2
TD-IDF			**BERT Feature-based**		**BERT Fine-tuned**	
Positive	96%	4%	53%	47%	82%	18%
Negative	100%	0%	53%	47%	87%	13%
Random Partition			**Category Partition**		**Polarity Partition**	
Positive	78%	22%	82%	18%	79%	21%
Negative	82%	18%	84%	16%	88%	12%

Table 3: Proportion of positive and negative reviews in each cluster

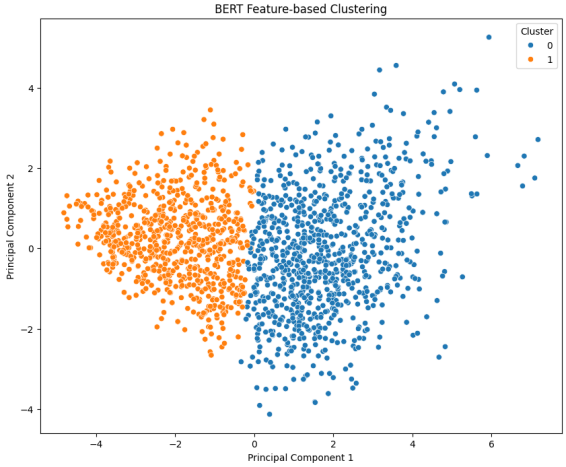
The top words for each clustering algorithm is highly similar and does not show signs of differentiation in polarity. Instead, it seems that words like "good", "service", "great", "excellent", and "staff" are ubiquitous top words between clusters. This is strange as the word embedding methods that account for polarity performs better in terms of the silhouette score. The BERT fine-tune model achieved a prediction accuracy of 0.93, but polarity is not reflected in the subsequent clustering of the word representation.

The lack of polarity distinction could be because k-means clustering picked up on other features not shown in results (rare words in text rather than top frequency words). Clustering error cannot fully explain poor polarity prediction as the graph below show strong clustering patterns [Figure 3].

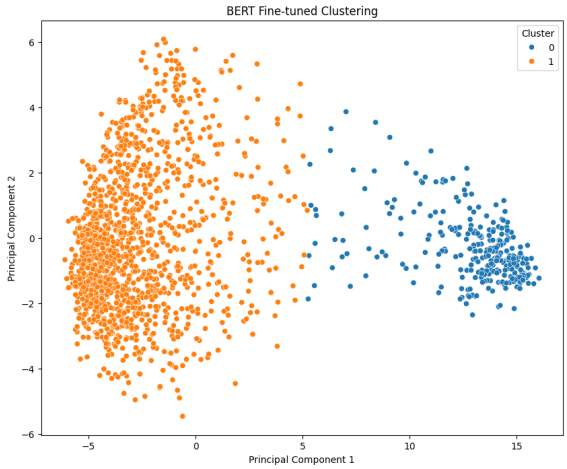
Comparing full-set (BERT fine-tuned) positive-negative proportions in clusters and the core-set clustering methods, we can further see that differentiation in polarity is not good. There is always one big cluster and one small cluster consisting of similar ratios of positive and negative reviews. It is inconclusive whether core-set algorithm can perform as good as full-set clustering when predicting polarity.



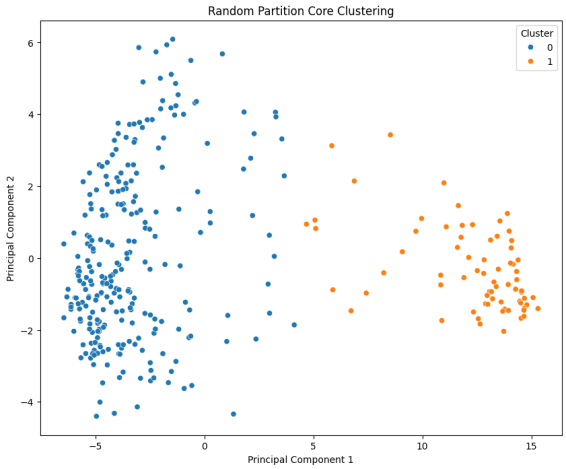
(a) TD-IDF Clustering



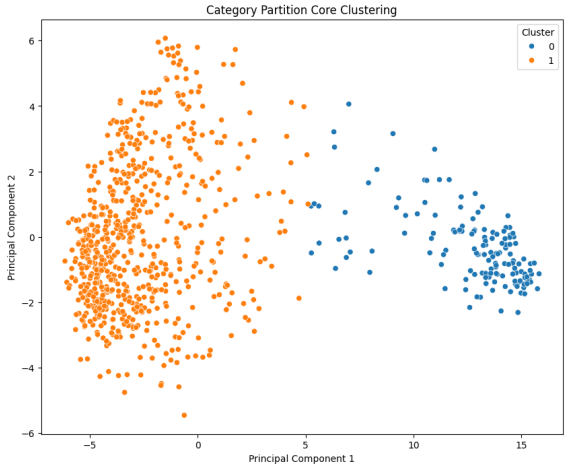
(b) BERT Feature-based Clustering



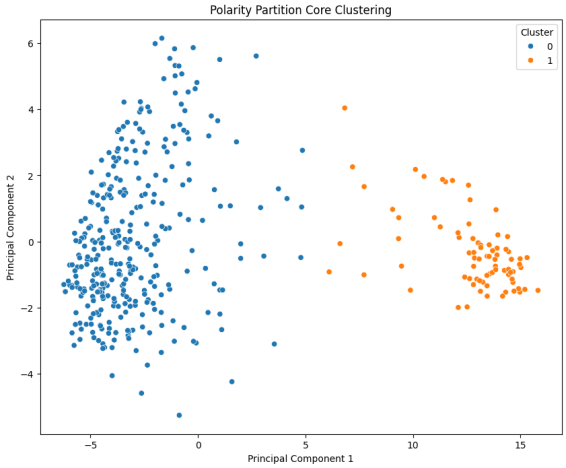
(c) BERT Fine-tuned Clustering



(d) Random Partition Coreset Clustering



(e) Category Partition Coreset Clustering



(f) Polarity Partition Coreset Clustering

Figure 3: Visualization of clustering algorithms

7 Conclusion and Further Experimentation

There is strong evidence that core-set clustering is an effective alternative to full-set clustering, and is particularly desirable when there is a large data set. It would require high-quality word embedding of the text. A previous attempt in core-set clustering with insufficiently encoded text yielded much worse results compared to full-set clustering. In addition, it is clear that the BERT fine tuning model had a significant positive impact on the performance of clustering. We can see from the graph that the clusters became more distinct and further apart from each other after fine-tuning [Figure 2c].

What is perplexing about this project is the lack of polarity distinction between the clusters despite the reasonable performance of the clustering algorithm. We will further try a few more experiments to find more information.

1. Setting k = 10

We set k = 10 for full-set clustering and random core-set clustering of the BERT fine-tuned word embedding. The resulting silhouette score were respectively 0.11 and 0.10.The performance of the clustering with a greater k decreased but the core-set clustering did not deteriorate in performance.

2. Recursive clustering on cluster 1 of BERT fine-tuned Clustering

We also tried to take the reviews from cluster 1 and recursively cluster them. Setting k=2, the silhouette score is 0.253, which is a decrease in performance compared to full-set clustering. This provides further evidence the BERT fine-tuning clustering had a good reason to be grouped into two clusters, but that reason was not polarity.

8 Further direction

Future investigations can try different clustering methods. One challenge for the clustering algorithm is to understand the nuance of the reviews where some sentences showed ambiguous or mixed sentiments, which can be decoded by researchers.

Furthermore, due to time constraints, we were not able to fully open the black box of clustering to examine the patterns that the unsupervised algorithm identified. There are remaining questions of models that account for polarity yielded better clustering performance (silhouette score) but cannot predict polarity. It is also unclear why the BERT fine-tuned model results did not give way to better clustering polarity prediction when it had an prediction accuracy of 0.93 internally. It would be beneficial to continue examining these questions.

References

[1] Alan Jeffares. *K-means: A complete introduction*. Nov. 2019. URL: https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c.

[2] Emil Hvitfeldt Silge and Julia. *Supervised machine learning for text analysis in R*. May 2022. URL: https://smltar.com/tokenization.html.

[3] Alvin Subakti, Hendri Murfi, and Nora Hariadi. "The performance of BERT as data representation of text clustering". In: *Journal of big Data* 9.1 (2022), p. 15.

[4] Cristina Zuheros et al. "Crowd Decision Making: Sparse Representation Guided by Sentiment Analysis for Leveraging the Wisdom of the Crowd". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53.1 (2023), pp. 369–379. DOI: 10.1109/TSMC.2022.3180938.